Amendments to the specification:

Please replace paragraph [0006] with the following amended paragraph:

[0006] ~~The main architectural limitation on the size of the register files is the number of bits available in the instruction to encode register specifiers.~~ Fixed length instructions and dense instruction encoding are key features of RISC architectures. For example, the PowerPC™ ISA (instruction set architecture) uses 32 bits to encode an instruction, with five bits allocated for each register specifier. In this architecture, instructions have between one and four register sources and destinations, such that there are only 12 bits remaining to encode the operation. Instruction encoding is tight, so that any attempt to increase the width of the register specifier fields would result either in longer instructions and code bloat, or in two-address instructions rather than the traditional RISC three-address instructions. We now discuss some known solutions to the problem of encoding register specifiers to increase the number of usable registers.

Please replace paragraph [0010] with the following amended paragraph:

[0010] U.S. Pat. No. 4,574,349 (issued to Rechtschaffen) interprets the register specifiers in an instruction as indices into an indirection table from which the actual register numbers are obtained. This solution suffers from the following drawbacks: (a) the indices to the indirection table are still limited to being log N bits wide--this makes the solution similar to a user managed register renaming scheme; (b) ~~they encode~~ Rechtschaffen encodes each register field independently, which requires either multiple indirection tables or multiple access ports into a shared indirection table~~, whereas we encode in a single entry of the indirection table all the register specifiers in an instruction~~. A related solution is proposed in the ISCA '93 paper by Kiyohara et al., titled "Register Connections: A New Approach to Adding Registers into Instruction Set Architectures."

Please add the following new paragraph after paragraph [0010]:

The Glossner Patent.

[0010.1] United States Patent 6,665,790 (issued to Glossner et al.) discloses a mechanism to address a vector register file through a pointer array. Each entry in the pointer array identifies at least one entry in the storage array (the vector register file). The main benefit of the pointer array is that it allows access to any set of registers, possibly a different set for each of the operands. In this solution, instructions still address registers (or pointer array entries) individually and not as a pattern, therefore suffering from the an instruction size limitation when trying to address a large register file.

Please delete the paragraph numbered [0011] and beginning with "The CodePack™ System."